IMPROVED METHOD FOR TESTING FOR THE PRESENCE OF FAULTS IN DIGITAL CIRCUITS

The present invention relates to a method of testing for the presence of faults in digital logic circuits. In particular, the invention also relates to a method for re-ordering "test vectors" to achieve an improved ordering which maximises the "fault coverage" on a digital circuit which is reached with a limited number of test vectors.

The testing of faults in digital circuits using test vectors is well known. This is typically done by applying a test vector, which is a set of binary values, to a digital circuit on either the primary inputs of the circuit or via a "scan chain" connecting the state elements, which are normally flip-flops, within the digital circuit and expected response values on the circuit primary inputs are captured within the state elements, prior to being accessed via the scan chain. The term fault coverage is defined as a measure of the number of faults within the circuit which an individual test vector or set of test vectors will detect. The common fault model is "stuck-at" faults which are represented by short-circuits to power or ground on the primary inputs or outputs of logic gates within the circuit. Fault coverage is measured using a fault simulator which determines whether the output response of the circuit is affected.

A large number of existing methods for generating test vectors are known and are published in the public domain. The best known is the Roth D Algorithm, originally published in 1966 and J.P. Roth "Diagnosis of Automota Failures: A Calculus and a Method", IBM Journal of Research and Development, Vol 10, No.4,pp.278-291, July, 1966.

Fig. 1 depicts an example of a test vector comprising two scan chains, which test vector is a list of binary values to be applied to the digital circuit under test via the primary input or via a scan chain. A scan chain is a method of serially loading values into the flip-flops in the digital circuit. This is a well known and widely published design technique. In the prior art example shown, two scan chains are depicted, although any number of scan chains may be used.

Existing methods of testing for faults on an integrated digital circuit suffer from the principal disadvantage that a very large amount of test vectors is required in order to provide a satisfactory fault coverage. This can take a relatively long time and requires a relatively large amount of memory and is a limiting factor in the design and testing of such circuits.

An object of the present invention is to obviate or mitigate at least one of the above-mentioned disadvantages.

This is achieved in the broadest sense by using randomly selected fault lists to select subsets of test vectors.

According to one aspect of the present invention, there is provided a method of maximising the fault coverage on an integrated digital circuit by re-ordering a number of test vectors for testing the digital circuit, said method comprising :

a) providing an initial set of test vectors T_0;

b) providing an original set of faults F_0;

c) selecting faults at random from the original fault list to form a sample fault list F_N;

d) forming a vector set T_{N-1} and simulating the vector set T_{N-1} against fault list F_N;

e) discarding any vector from the vector set T_{N-1} which does not detect any faults;

f) saving the remaining vectors as vector set T_N;

g) repeating the above steps c) to f) N times with N having a value of 1 to M so that at the end of M steps, test vectors T_1 to T_M are saved;

h) removing duplicate vector patterns in each vector set T_N; and

i) initialising the final vector set and appending vector sets V_M through V_0 to produce a final vector set T_F.

Preferably, in step g) M is 10 and steps c) to f) are therefore repeated ten times.

Preferably, the method of removing duplicate vector patterns is achieved by :

j) copying the original fault list F_O to provide a secondary fault list G_N;

k) fault simulating vector set T_N against G_N and deleting any vectors which find no faults;

l) saving the resulting vectors as vector set V_N and saving the list of undetected faults as list G_{N-1};

m) repeating steps k) and l) M+1 times with N having values M to 0;

These and other aspects of the invention will become apparent from the following description, when taken in combination with the accompanying drawings, in which :

Fig. 2 is a flow chart of the steps involved in the method of re-ordering test vectors for maximising the fault coverage on a digital circuit;

Fig. 3 is a legend to terms used in the flow chart of Fig. 2, and

Fig. 4 is an example of a graph of fault coverage showing the number of faults detected against the number of vectors for the original vectors and vectors after using the new method/algorithm.

Reference is first made to Figs. 2 and 3 of the drawings which depict a flow chart of a sequence of steps involved in re-ordering test vectors to provide a test vector generation pattern for maximising the fault coverage on an integrated digital circuit using a limited number of test vectors.

In stage 1, (steps a) to f)) an initial set of vectors are provided (step a) and these are copied to form a set of test vectors T_O to place these vectors in a near-optimal order for detecting a set of faults F_O. Stage 1 has two principal steps. In the first step (step 10), a list of faults is selected at random with a probability of 2^{-N} from the original fault list F_O to form a sample or subset fault list F_N. The second major step (step 12) in stage 1 is fault simulating the vector set T_{N-1} repeatedly against the fault list F_N and then discarding any vector which does not detect any faults. In step 12, the ordering of individual vectors within the vector set are alternately reversed and randomised. The resulting vector set, that is those that do result in the finding of faults, is saved as vector set T_N.

The two major steps 10 and 12 are repeated M times where M is 10 in this example.

At the completion of the tenth step 12, there are ten sets of vectors saved, T_1 to T_{10}.

At the end of the stage 1, the method involves stage 2 wherein duplicate vector patterns are removed from the vector pattern list T_N. In this case the original fault list F_O is again copied and denoted as fault list G_N (step 14). In the next step in stage 2, step 16, the vector set T_N is fault simulated against the fault list G_N and any vectors which result in no faults being found are deleted. After the fault simulation, the resulting vector set is saved as V_N and the list of undetected faults is saved as G_{N+1}. Stage 2 is repeated M plus 1 times, with N taking values M down to zero where M is 10.

The final stage in the methodology is stage 3 in which the final vector set is initialised (step 18) and vector sets V_M to V_O are appended together to produce a final vector set T_F (step 20).

The re-ordering of the generated test vectors in the way described above allows a digital integrated circuit to be tested much quicker than with the prior art test vector ordering, such that a digital circuit can be tested in typically one tenth of the time using the prior art re-ordering test vector set. This means that much less memory is required and the design process is speeded up, resulting in a considerable economic benefit.

Fig. 4 depicts a graph of faults detected against a number of vectors. It will be seen that a larger number of faults are detected for vectors re-ordered after the new re-ordering method compared to an original number of vectors where there is less than about 700 vectors used. This increase is most dramatic for a lower

number of vectors, such that this re-ordering algorithm maximises the fault coverage for a lower number of vectors when testing an integrated digital circuit.

Various modifications may be made to the re-ordering methodology hereinbefore described, without departing from the scope of the invention. For example, the repetition of each stage may take values in excess of or less than 10, although this will have an effect on the time taken to test the digital circuit. In addition, the probability factor of X^{-N} in this example may be varied. Typically, the value of X is 2. Increasing the value of X from 2 will reduce the time taken for the re-ordering but will reduce the quality of result and decreasing the value X from 2 will result in a longer time for re-ordering but will be more accurate. In addition, the duplicate vector patterns in the vector sets are removable by an alternative method wherein a text search is conducted through the list of files of vector patterns to look for identical vector patterns and once the identical vector patterns have been identified they are deleted. It will be appreciated that after re-ordering the vectors, a larger number of faults are detected by any size of subset of the original vectors than by the original vectors themselves and, as indicated above, this significantly reduces the test time for the digital circuit and requires less memory capacity in the test apparatus, resulting in a more effective and more efficient test system.